

R and Datamining with R

Philipp Kinschel, Anton Pölzleitner

Universität Salzburg, Fachbereich Computerwissenschaften

2. Juni 2010

Inhalt

- R-Project
- Funktionsüberblick von R
- Datamining
- Dataminingbeispiel mit R

R-Project

- <http://www.r-project.org/>
- Funktionelle Sprache für statistische Berechnungen und Auswertungen
- Objektorientiert
- Opensource, Plattformen: Linux, OS X, Windows
- Dialekt von S

Objekte

- Variablen, Funktionen, Daten im Speicher
- Zuweisung durch " $<-$ " oder " $->$ "
- Solange im Speicher bis es gelöscht wird
- `ls()` oder `objects()` fuer Liste aller Objekte

Beispiel

```
> z <- 5  
> w <- z2  
> w  
[1]25  
> i <- (z * 2 + 45)/2  
> i  
[1]27.5
```

Vektoren

- Elementarste Datenobjekt
 - ▶ Special Property "mode" & "length"
 - ▶ mode = Typ der Daten
 - ▶ length = Anzahl der Elemente
- Alle Elemente eines Vectors immer vom gleichen Typ
- "NA" repräsentiert unbekannten Wert

Beispiel

```
> v <- c(4,7,23.5,76.2,80)
> v
[1] 4.0 7.0 23.5 76.2 80.0
> length(v)
[1] 5
> mode(v)
[1] "numeric"
```

Vektorisierung

- Einer der mächtigsten Aspekte von R
- Alle Funktionen von R sind auf Vektoren anwendbar
- Sowie Vektorarithmetik

Beispiel

```
> v1 <- c(4, 6, 8, 24)
```

```
> v2 <- c(10, 2, 4)
```

```
> v1 + v2
```

```
[1] 14 8 12 34
```

Warning message:

longer object length

is not a multiple of shorter object length in: v1 + v2

Faktoren

- Einfacher Weg mit nominalen (kategorischen) Daten umzugehen
- Jeder Faktor besteht aus ein oder mehreren "level"
- `table()` fuer Crosstabulation

Beispiel

```
> g <- c('f','m','m','m','f','m','f','m','f','f')
> g
[1] "f" "m" "m" "m" "f" "m" "f" "m" "f" "f"
> g <- factor(g)
> g
[1] f m m m f m f m f f
Levels: f m

> table(g)
g
f m
5 5
```

Sequenzen generieren

- Viele Werkzeuge um verschiedene Sequenzen zu erzeugen
- ":", seq(), rep(), gl(), rnorm(), rt()

Beispiel

```
> seq(-4,1,0.5)
[1] -4.0 -3.5 -3.0 -2.5 -2.0 -1.5 -1.0 -0.5 0.0 0.5 1.0
> seq(length=10,from=-2,by=.2)
[1] -2.0 -1.8 -1.6 -1.4 -1.2 -1.0 -0.8 -0.6 -0.4 -0.2
> rep('hi',3)
[1] "hi" "hi" "hi"
> rep(1 : 3, 2)
[1] 1 2 3 1 2 3
> gl(3,5)
[1] 1 1 1 1 1 2 2 2 2 2 3 3 3 3 3
Levels: 1 2 3
```

Matrizen und Arrays

- Datenelemente mit mehr als einer Dimension
- Matrix = Spezialfall mit zwei einfachen Dimensionen
- `matrix()` um aus Vektoren Matrizen zu erstellen
- `cbind()`, `rbind()` um Vektoren, Matrizen zu verknuepfen

Listen

- Geordnete Sammlung von Objekten, genannt "components"
- "type", "mode", "length" müssen nicht gleich sein
- "components" immer nummeriert, Bezeichner optional

Beispiel

```
> my.lst <- list(stud.id=34453,  
+               stud.name=" John",  
+               stud.marks=c(14.3,12,15,19))  
  
> my.lst  
$stud.id  
[1] 34453  
$stud.name  
[1] " John"  
$stud.marks  
[1] 14.3 12.0 15.0 19.0
```

Data frames

- Ist eine Matrix mit Bezeichner fuer Spalten

Beispiel

```
> my.dataset <- data.frame(site=c('A','B','A','A','B'),  
+ season=c('Winter','Summer','Summer','Spring','Fall'),  
+ pH = c(7.4,6.3,8.6,7.2,8.9))
```

```
> my.dataset
```

```
site season pH
```

```
1 A Winter 7.4
```

```
2 B Summer 6.3
```

```
3 A Summer 8.6
```

```
4 A Spring 7.2
```

```
5 B Fall 8.9
```

```
> my.dataset[3,2]
```

```
[1] Summer
```

```
Levels: Fall Spring Summer Winter
```

Funktionen erstellen

- Auf die selbe Weise angewendet wie Daten
- Aufbau: `function(<list of parameters>) {<list of R instructions>}`
- Bedingung: `if(<list of conditions>) {<list of R instructions>}`
- Schleife: `for(i in 1:10) {<list of R instructions>}`

Beispiel

```
> se <- function(x) {  
+ v <- var(x)  
+ n <- length(x)  
+ return(sqrt(v/n))  
+ }  
> se(c(45,2,3,5,76,2,4))  
[1] 11.10310
```

Sessionmanagement

- Command line fuer komplexere bzw. groeßere Aufgaben nicht geeignet
- `source(<Filename>)` laedt alle Befehle, Funktionen, Anweisungen der Datei
- `save()` speichern einzelner Funktionen, Objekte
- `save.image()` speichert kompletten aktuellen workspace

Beispiel

```
> save(f,my.dataset,file='mysession.R')
```

Data Import/Export

- Üblichste Variante sind Daten aus Textfiles
- XML-Unterstützung zur Erleichterung beim Import/Export
- Binary-Dateien von S-PLUS, SAS, SPSS, Stata, Systat...
- DB-Unterstützung von Microsoft SQL Server, Access, MySQL, PostgreSQL, Oracle, IBM DB2, SQLite

Beispiel

```
> student <- read.table('Analysis.txt',  
+                       header=F,  
+                       dec='.',  
+                       col.names=c('id', 'name', 'marks'),  
+                       na.strings=c('XXXXXXXX'))
```

wichtige Datenfunktionen(1)

- `sum(x)`
 - ▶ Sums the elements in `x`
- `prod(x)`
 - ▶ Product of the elements in `x`
- `max(x)`
 - ▶ Maximum element in `x`
- `min(x)`
 - ▶ Minimum element in `x`
- `range(x)`
 - ▶ Range (min to max) of elements in `x`
- `length(x)`
 - ▶ Number of elements in `x`
- `mean(x)`
 - ▶ Mean (average value) of elements in `x`.
- `median(x)`
 - ▶ Median (middle value) of elements in `x`

wichtige Datenfunktionen(2)

- `var(x)`
 - ▶ Variance of elements in `x`
- `sd(x)`
 - ▶ Standard deviation of element in `x`
- `cor(x,y)`
 - ▶ Correlation between `x` and `y`
- `quantile(x,p)`
 - ▶ The `p`th quantile of `x`
- `cov(x,y)`
 - ▶ Covariance between `x` and `y`

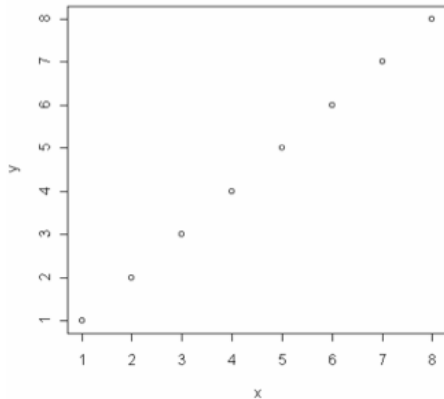
Diagrammfunktionen(1)

- `pie(x)`
 - ▶ Circular pie chart
- `hist(x)`
 - ▶ Histogram of the frequencies of x
- `barplot(x)`
 - ▶ Histogram of the values of x
- `stripchart(x)`
 - ▶ Plots values of x along a line
- `dotchart(x)`
 - ▶ Cleveland dot plot

Diagrammfunktionen(2)

- `pairs(x)`
 - ▶ For a matrix `x`, plots all bivariate pairs
- `plot.ts(x)`
 - ▶ Plot of `x` with respect to time (index values of the vector unless specified)
- `contour(x,y,z)`
 - ▶ Contour plot of vectors `x` and `y`, `z` must be a matrix of dimension `rows=x` and `columns=y`
- `image(x,y,z)`
 - ▶ Same as contour plot but uses colors instead of lines

Boxplot-Beispiel(1)

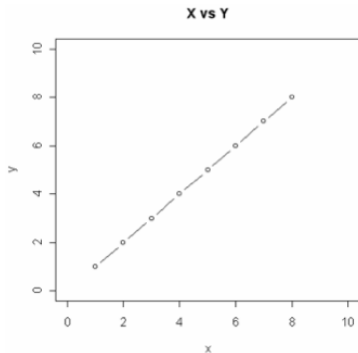


Code

```
> x<-c(1,2,3,4,5,6,7,8)
> y<-c(1,2,3,4,5,6,7,8)
> plot(x,y)
```

Boxplot-Beispiel(2)

- Wertebereich, Achsenbezeichnung, Titel

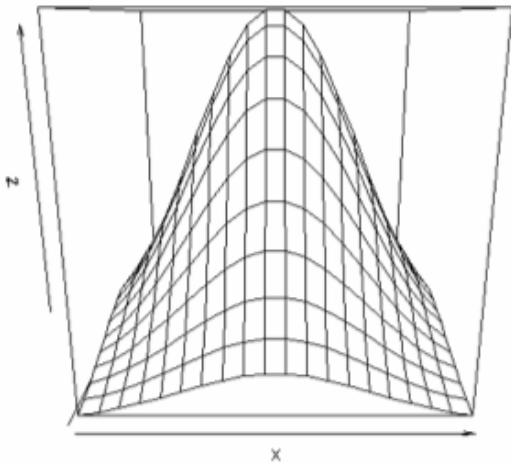


Code

...

```
> plot(x,y,xlim=range(0:10),ylim=range(0:10),type='b',main="X vs Y")
```

3D contour plot



Datamining

- Sammlung von Daten
 - ▶ Versandhandel, Bank, Versicherung, Naturwissenschaft, ...
- Daten bieten Fülle an wertvollen Informationen
- Daten liegen Brach im System
- Herleitung des Begriffs: Datamining
 - ▶ Aus "Datenberg" wertvolle Informationen aggregieren
- Aggregation von Daten mittels statistischer Auswertung

Datamining

- Datenaufbereitung
- Überprüfen
- Ersetzen von fehlenden Daten
- Transformation der Daten
- Prognosen

Algenmessung

- Motivation: Wasserqualitätssicherung
 - ▶ Herausfinden wann Algen blühen (werden)
- Daten
 - ▶ Wasserproben von versch. Gewässern Europas innerhalb eines Jahres in verschieden langen Abständen

Messdaten

- chemische Eigenschaften
- Häufigkeit von 7 verschiedene schädlichen Algen
 - ▶ max pH Wert, min. Sauerstoffgehalt, Chloride, Phosphate,
- Jahreszeit
- Flussgröße
- Flussgeschwindigkeit

Einlesen

Code

```
> algae <- read.table('Analysis.txt', header=F, dec='.',  
+ col.names=c('season','size','speed','mxPH','mnO2','Cl',  
'NO3','NH4','oPO4','PO4','Chla','a1','a2','a3','a4','a5','a6','a7'),  
+ na.strings=c('XXXXXXXX'))
```

- header=F - erste Zeile ignorieren
- dec='.' - Kommazeichen
- col.names - Vektorenbezeichnung
- na.strings - unbekannter Wert

Übersicht

Code

```
> summary(algae)
```

season	size	speed	mxPH	mn02
autumn:40	large :45	high :84	Min. :5.600	Min. : 1.500
spring:53	medium:84	low :33	1st Qu.:7.700	1st Qu.: 7.725
summer:45	small :71	medium:83	Median :8.060	Median : 9.800
winter:62			Mean :8.012	Mean : 9.118
			3rd Qu.:8.400	3rd Qu.:10.800
			Max. :9.700	Max. :13.400
			NA's :1.000	NA's : 2.000

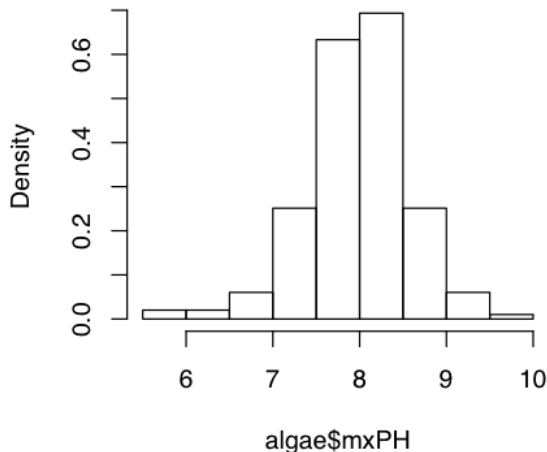
C1	NO3	NH4	oP04
Min. : 0.222	Min. : 0.050	Min. : 5.00	Min. : 1.00
1st Qu.: 10.981	1st Qu.: 1.296	1st Qu.: 38.33	1st Qu.: 15.70
Median : 32.730	Median : 2.675	Median : 103.17	Median : 40.15
Mean : 43.636	Mean : 3.282	Mean : 501.30	Mean : 73.59
3rd Qu.: 57.824	3rd Qu.: 4.446	3rd Qu.: 226.95	3rd Qu.: 99.33
Max. :391.500	Max. :45.650	Max. :24064.00	Max. :564.60
NA's : 10.000	NA's : 2.000	NA's : 2.00	NA's : 2.00

...

PH-Wert als Histogramm

Code

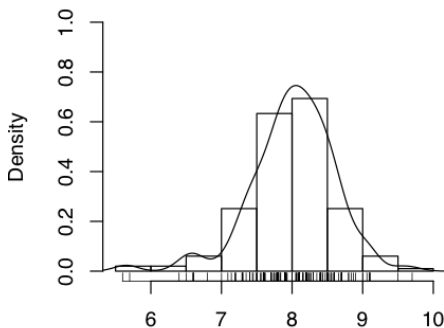
```
> hist(algae$mxPH, prob=T)
```



PH-Wert als Histogramm

Code

```
> hist(algae$mxPH, prob=T, xlab="",  
+ main='Histogram of maximum pH value',ylim=0:1)  
> lines(density(algae$mxPH,na.rm=T))  
> rug(jitter(algae$mxPH))
```



unvollständige Datensätze

- Datensätze entfernen
 - ▶ wenn nur wenige Daten fehlen
- befüllen - mit häufigsten Wert
 - ▶ mit Mean (Normalverteilung)
 - ▶ mit Median wenn Daten stärker variieren
- befüllen - Wert ermitteln durch Korrelation zw. Werten
- befüllen - Wert ermitteln durch Fallvergleiche

Literaturverzeichnis

- Data Mining with R: learning by case studies by Luis Torgo, LIACC-FEP, University of Porto
- R: A language for data analysis and graphics by R. Ihaka and R. Gentleman
- Statistics Using R with Biological Examples by Kim Seefeld, Ernst Linder, University of New Hampshire
- R Data Import/Export by R Development Core Team
- <http://www.r-project.org/>